Int. J. Math. Model. Comput., 2026; 16(2): 138-152

International Journal of Mathematical Modelling & Computations (IJM^{2}C)

https://doi.org/10.57647/ijm2c.2026.1602.12

Research Article

Efficient Multi-Path CNN Architecture for Anomaly Detection in Control Charts with Imbalanced Data

Amir Azar, Sadigh Raissi*, Paria soleimani, Shahrooz Bamdad

Department of Industrial Engineering, ST.C., Islamic Azad University, Tehran, Iran

***Corresponding author:** raissi@azad.ac.ir

Original Research:

Received:
15 August 2025

Revised: :
10 September 2025

Accepted:
15 November 2025

Published Online:
22 November 2025

Published in issue:
30 June 2026

© 2026 the Author(s). Published by the OICC Press under the terms of the CC BY 4.0, Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Abstract

Control Chart Pattern Recognition (CCPR) is critical for early anomaly detection in industrial processes, yet data imbalance poses a significant challenge, leading to biased models that overlook rare but critical anomalies. This study introduces a novel Multi-Path SMOTE-CNN framework to address this issue, enhancing classification performance under moderate (1:20) and severe (1:200) imbalance ratios. The methodology integrates univariate and dual-channel Synthetic Minority Over-sampling Technique (SMOTE) extensions with a multi-path Convolutional Neural Network (CNN) that processes raw time-series and recurrence plot images in parallel, extracting complementary temporal and spatial features. The model was evaluated on simulated datasets and the real-world Wafer dataset using accuracy, sensitivity, specificity, and G-mean metrics. Results demonstrate significant improvements over the baseline CSCNN model, achieving G-mean values up to 0.9928 for cyclic patterns under severe imbalance and 0.9987 on the Wafer dataset. The MP-SMOTE-CNN's ability to handle extreme imbalance, reduce manual feature engineering, and ensure computational efficiency highlights its potential for real-time industrial applications. This work advances CCPR by offering a robust, scalable solution for anomaly detection, with implications for quality control in manufacturing and beyond.

Keywords: Control Chart Pattern Recognition (CCPR); Convolutional Neural Network; Imbalanced Data; Synthetic Minority Over-sampling Technique (SMOTE); Anomaly Detection; Recurrence Plot

Cite this article: Azar, A., Raissi, S., Soleimani, P. & Bamdad, Sh., (2026). Efficient Multi-Path CNN Architecture for Anomaly Detection in Control Charts with Imbalanced Data, Int. J. Math. Model. Comput., 16(2): 138-152. https://doi.org/10.57647/ijm2c.2026.1602.12

1. Introduction

The recognition of control chart patterns (CCPR) plays a crucial role within Statistical Process Control (SPC), allowing for the prompt identification of unusual patterns—including shifts, trends, or repetitive cycles—in industrial operations, which may signal deviations that threaten product quality and the stability of the process. Timely identification of these patterns is essential for diagnosing root causes and implementing corrective measures to ensure operational efficiency. However, a critical challenge in applying machine learning to CCPR is data imbalance, where abnormal patterns (minority class) are significantly underrepresented compared to normal patterns (majority class). This imbalance severely impacts neural network performance, as models often exhibit a bias toward the dominant class, which in turn hampers the accurate identification of rare but critical anomalies, reduced sensitivity, and compromised reliability in industrial applications [1].

Data imbalance poses a major challenge in neural network-based CCPR due to its prevalence in real-world industrial datasets, where abnormal events are inherently

rare but have significant consequences. Without addressing this issue, models may become overly fitted to the dominant class, leading to elevated false-negative occurrences in minority categories and reducing the overall efficiency of SPC. The literature highlights four primary methods to mitigate data imbalance in neural networks: (1) Data resampling involves methods such as increasing the representation of minority classes (for example, the Synthetic Minority Over-sampling Technique, SMOTE [2] or reducing the number of samples in majority classes to achieve a balanced dataset; (2) Class Weighting, where higher weights are assigned to minority classes during training to prioritize their learning; (3) Ensemble learning, which integrates several models, is employed to boost performance when working with imbalanced datasets. Additionally, customized architectures—such as adapted Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models—are specifically developed to address the challenges of imbalanced data [3]. Within this context, SMOTE has become a key method, primarily for its capacity to produce artificial minority instances, thereby improving the model's generalization capability without merely replicating existing data.

Recent research has significantly advanced methods for handling class imbalance in CCPR and related domains. Early work by Farsi et al. [4] demonstrated the potential of chaotic neural networks for pattern recognition, laying the groundwork for advanced architectures. Building on this, Buda et al. [5] highlighted the vulnerability of CNNs to imbalanced data, motivating the design of bias-mitigation strategies. This concern was further addressed by Fuqua and Razzaghi [6], who proposed a cost-sensitive CNN for imbalanced CCPR, achieving robust results in simulated and real datasets. Complementary advances in semiconductor manufacturing were provided by Wang et al. [7] through a VAE-enhanced deep learning framework that both augmented wafer defect maps and improved interpretability.

A second cluster of research has focused on oversampling and feature engineering to mitigate imbalance. Swana et al. [8] showed that combining SMOTE with Tomek Link resampling produced superior fault classification in electrical machines, while Duan et al. [9] introduced MeanRadius-SMOTE to avoid noisy synthetic data, outperforming traditional SMOTE variants. For CCPR specifically, Xue et al. [10] developed a multi-feature fusion CNN (MFF-CNN) that, together with SMOTE, outperformed conventional classifiers in both single and mixed patterns. Similarly, Cheng et al. [11] demonstrated that a multi-channel CNN combining raw data and recurrence plots improved multivariate CCP recognition by up to 10%. In broader industrial applications, Naser et al. [12] proposed an AI-based operational excellence model, highlighting AI's transformative role in predictive analytics and pattern recognition for manufacturing sectors.

Most recently, new architectures have targeted subtle shifts, variable windows, and dynamic data. Li et al. [13] introduced a multi-scale weighted ordinal pattern ensemble classifier (WOP-EC) that excelled in detecting small and large shifts, achieving up to 99.88% accuracy. Chu et al. [14] combined SMOTE with smooth shift processing and stacking ensembles, yielding 97.31% accuracy in dimensional quality control CCPR. Pushing further, Xue et al. [15] integrated CNNs with convolutional block attention modules and multi-feature fusion to achieve 99.73% accuracy and enable online monitoring under imbalanced dynamic conditions. Finally, Zan et al. [16] extended CCPR to variable-length charts with a sliding window SECNN-BiLSTM model, validated in both simulations and cloud-edge industrial applications. Collectively, these studies show a clear evolution from foundational neural methods toward highly specialized, hybrid, and adaptive architectures designed to tackle imbalance and variability in modern CCPR contexts.

Despite these advancements, most studies focus on moderately imbalanced datasets, with limited exploration of severely imbalanced scenarios, which are common in industrial applications where abnormal patterns constitute a small fraction of data. Furthermore, the integration of SMOTE with multi-path CNN architectures to leverage diverse data representations (e.g., raw time-series and recurrence plots) remains underexplored, limiting the ability to capture complementary features for robust classification. This study proposes a novel Multi-Path SMOTE-CNN (MP-SMOTE-CNN) framework to address CCPR under both moderate and severe data imbalance. By combining univariate and dual-channel SMOTE extensions with a multi-path CNN architecture, the proposed approach generates high-quality synthetic samples and extracts complementary features from raw time-series and recurrence plots. This dual-channel strategy enhances classification accuracy and sensitivity for minority classes, offering a scalable solution for real-time anomaly detection. The performance of MP-SMOTE-CNN is evaluated on simulated datasets with imbalance ratios of 1:20 and 1:200 and the real-world Wafer dataset, demonstrating significant improvements over existing methods, including the baseline CSCNN model [6]. Interested readers can refer to Table 1 in the article's appendix to understand the strengths and differences of research efforts in this field.

10.57647/ijm2c.2026.1602.12

2. Research Basis

a) Machine learning: In the Control Chart Pattern Recognition (CCPR) problem, input data are presented as time series. To this end, we consider a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, comprising *n* univariate time series. In this dataset, if T is a positive integer, $x_i \in R^T$ represents the *i-th* sample, and $y_i \in \{0,1\}$ denotes the class label it corresponds to. In time-series analysis, the parameter *T*—often called the window length—specifies how much data is considered for the evaluation of each sample.

Within supervised learning frameworks, the main aim is to determine the most suitable parameter settings that reduce the cost function, representing the model's prediction error. In the case of Convolutional Neural Networks (CNNs)—an enhanced variant of conventional artificial neural networks—this entails adjusting parameters such as ω *and* b, which establish the decision boundary.

$$E(w,b) = \frac{1}{n}\sum_{i=1}^{n} l\big(y_i, \hat{y}_i(w,b)\big) \quad (1)$$

Given $x_i, i \in \{1,\dots,n\}$ as a sample from the dataset and y_i as the target output, the predicted value by the algorithm is denoted as $\hat{y}_i(w,b)$, where $l(.)$ is the cost function. The symbols ω and b denote the collection of weight and bias parameters within the network. For brevity in notation, the predicted value $\hat{y}_i(w,b)$ will be referred to simply as $\hat{y}_i$.

In artificial neural networks, various types of cost functions $l(.)$ are employed, with the most common being Mean Squared Error (MSE) and Cross-Entropy (CE). The CE cost function, also known as "Softmax Log Loss," is particularly reliable in classification tasks, especially in image recognition [17]. In contrast, for processing time series data, the MSE cost function is recommended [6]. Given that the input data in this study are time series, the MSE cost function is adopted, which calculates the square of the difference between the true output and the value estimated by the algorithm.

$$E(w,b) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (2)$$

b) Deep Learning via Convolutional Neural Network: Artificial Neural Networks (ANNs) represent one of the most prevalent types of machine learning models, structured in multiple layers made up of interconnected neurons. Each neuron receives inputs, applies a mathematical transformation dependent on trainable weights and a bias term, and produces an output. This process also involves a nonlinear transformation performed by an activation function. In such networks, every neuron within a layer maintains full connectivity with the outputs from the preceding layer. To optimize the model and reduce prediction error, these networks employ the backpropagation algorithm, which adjusts weights and biases to enhance prediction accuracy [6].

$$[o]_{f,l} = g\big(b + \sum_{h=1}^{r}\sum_{j=1}^{q} w_{h,j}^{f} * z_{h,r+j-1}^{i}\big) \quad (3)$$

Convolutional Neural Networks (CNNs), which are a distinct category of artificial neural networks, are intended for extracting critical features. This capability is achieved by imposing constraints on weights and biases in specific layers called convolutional layers and by performing down sampling operations in pooling layers. Consequently, these networks contain three main kinds of layers: convolutional layers responsible for feature extraction, pooling layers aimed at reducing dimensionality, and fully connected layers that function in a manner comparable to conventional ANNs. The typical structure of a CNN consists of multiple stacks of convolutional and pooling layers, which ultimately feed compressed data and extracted features into fully connected layers. In this process, raw input data are first passed to the convolutional layer, which generates a representation of local data features. Unlike standard ANNs, convolutional layers consist of a set of identical neurons referred to as filters (kernels), in which each neuron links exclusively to a limited portion of the input data. These constraints reduce the number of parameters required for training the network. Notably, every neuron within the convolutional layer continues to utilize an activation function comparable to those used in ANNs. The details of the proposed MP-SMOTE-CNN model will be introduced based on its structure in the following.

A convolutional layer is composed of multiple filters *F*, a bias parameter *b*, and an activation function *g*—each unique to the layer and adjusted during training. Although using a separate index to denote layers may seem logical, for notational simplicity, such indices are intentionally omitted. Assume that $z^i \in \mathbb{R}^{r\times p}$ represents the *i*-th input to a given convolutional layer is processed by filters $f = 1, \dots, F$ where each filter is represented as a matrix $\omega^f \in \mathbb{R}^{r\times q}$ with q being smaller than p. Here, p denotes the input length prior to the convolutional operation, and q represents the length of the filter. Accordingly, the output of this convolutional layer is a matrix of size $(F \times (p - q + 1))$ with elements given by:

where $f = 1, \dots, F$ and $l = 1, \dots, p - q + 1$. The chosen activation function is the Rectified Linear Unit (ReLU), which can be expressed as follows:

10.57647/ijm2c.2026.1602.12

$$ReLU: g(\xi) = max(0, \xi) \quad (4)$$

This activation function is utilized across every convolutional layer because it offers a considerably faster training process compared to alternative activation functions [18].

For a univariate time series, applying a single filter yields another univariate time series but with a shorter length. When multiple filters (F) are employed, the outcome becomes a multivariate time series formed by stacking F separate univariate sequences. In this scenario, the first convolutional layer processes a univariate sequence using F filters of dimensions $1 \times q$, assuming $r = 1$ for the original input data. The output generated differs from the one-dimensional input, producing a matrix of size $F \times (T - q + 1)$, with p representing T for the initial dataset. In deeper layers, the filters evolve from one-dimensional forms to structures whose height matches the input size of the corresponding convolutional layer. This characteristic enables the MP-SMOTE-CNN to extract diverse features by applying varied filters across different layers, which are beneficial for improving accuracy in the final classification.

When a single filter is applied to a univariate time series, the outcome remains a univariate series but with reduced length. In contrast, employing multiple filters (F) transforms it into a multivariate time series, produced by stacking F possibly distinct univariate series. In this setup, the initial convolutional layer processes a univariate sequence through F filters of size $1 \times q$, assuming $r = 1$ for the original input data. The resulting output is a matrix of size $F \times (T - q + 1)$, where p corresponds to T in the case of raw input. In the following layers, the filters transition from one-dimensional to shapes with heights matching the input dimensions of each convolutional layer. By using various filters across layers, the MP-SMOTE-CNN is capable of extracting diverse and informative features that contribute to improved final classification performance.

Note 1. The extended form of Equation (3) is derived by substituting $z^i_{h,r+j-1}$ with $z^i_{h,r+j-s_c}$, where s_c denotes the stride parameter. When $s_c = 1$, a filter of dimensions $r \times q$ is initially applied to the time indices $1, \dots, q$ of the input, followed by $2, \dots, q+1$, and continuing in the same manner. For a general s_c, the filter begins at $1, \dots, q$; and is then shifted to $1 + s_c, \dots, q + s_c$. Increasing the stride value reduces the output size. In this work, a stride of $s_c = 1$ is is used for every convolutional layer.

Within CNN architectures, the result produced by a convolutional layer can be sent directly to another convolutional layer or to a pooling layer. Pooling layers serve to decrease data dimensionality and capture more abstract features by performing downsampling, which involves applying an aggregation function to the input. This process is performed using a sliding window that computes the output value based on an aggregation function. This function may involve "maximum" or "average" operations, but Empirically, max pooling has shown better capability in capturing sparse features than average pooling [19]. In the present work, max pooling is applied at all stages except for the last pooling layer, where average pooling is utilized.

Let $u^i \in \mathbb{R}^{r \times p}$ represent the data entering a pooling layer, with k indicating the size of the sliding window applied for input aggregation. The resulting output from the pooling layer forms a matrix of dimensions $(r \times (p - k + 1))$ whose elements are defined as follows:

$$[o]_{l,v} = max\{u^i_{l,j}: \ j = v, \dots, v + p - k + 1\} \quad (5)$$

where $l = 1, \dots, r$ and $v = 1, \dots, p - k + 1$. Substituting the "max" function in Equation (5), with the "avg" function results in the creation of an average pooling layer. In CNN-based time series applications, typical pooling filter dimensions are $k = 2$ and $k = 3$. Increasing the pooling size beyond these values generally produces notably inferior outcomes [6].

Note 2. Analogous to Equation (3), the generalized form of Equation (5) is derived by altering j so that it assumes values from the set $v, \dots, v + p - k + s_p$, with s_p denoting the stride parameter specific to pooling. This parameter influences the pooling process in a way similar to how the stride parameter functions within a convolutional layer.In CNNs, convolutional and pooling layers are stacked sequentially until a representative two-dimensional feature map is obtained. Following feature extraction, this map is flattened and transformed into a one-dimensional vector, which serves as input to a Multilayer Perceptron (MLP). The MLP, functioning as a conventional feedforward neural network, processes the input to generate a single feature vector and conducts binary classification via a sigmoid activation function. Similar to other neural architectures, CNNs utilize gradient-based backpropagation to fine-tune weights and biases, aiming to reduce the error function.

c. Recurrence Plot: A recurrence plot is a technique used to represent time series data visually by reconstructing the phase space—also known as the trajectory—of the series. The initial step in this process involves reconstructing the phase space of the data. Depending on the choice of threshold, the output image can be binary or grayscale. In this study, the approach proposed by Cheng et al., [11] is followed, and a

threshold-free technique is employed to extract more information from the recurrence plot images. Specifically, recurrence plots were generated using the pyts.image.RecurrencePlot transformer in Python. As no additional arguments were specified, the default settings were applied: embedding dimension = 1, time delay = 1, Euclidean distance metric, and no thresholding (i.e., threshold-free representation). This configuration ensures that recurrence plots capture temporal dependencies without information loss.

d. MP-SMOTE-CNN: SMOTE, short for Synthetic Minority Over-sampling Technique, is a data preprocessing method extensively applied in diverse fields and has formed the basis for many approaches aimed at tackling class imbalance. This technique has played a significant role in establishing new paradigms for supervised learning, such as multi-label classification. Unlike traditional methods that perform oversampling through replication, SMOTE balances minority classes by generating new synthetic samples [3]. Such synthetic samples are generated by leveraging feature similarities from the original minority class instances, instead of considering the complete data space, which in turn improves the model's ability to generalize. By expanding decision regions and preventing overfitting, SMOTE is recognized as one of the most effective sampling algorithms in data preprocessing for machine learning and data mining. In this method, For every instance belonging to the minority class, the algorithm locates its k closest neighbors within the same class and produces synthetic instances by interpolating between these points. Specifically, for a subset of the minority class $S_{min} \in S$, consider the k-nearest neighbors for each example $x_i \in S_{min}$; the k nearest neighbors of x_i are the k points in the n-dimensional feature space x that have the shortest Euclidean distance to x_i.

To create a new synthetic instance x_{new}, a random selection is made from the k nearest neighbors of x_i The vector difference between this chosen neighbor and the original instance is then computed and scaled by a randomly generated number between [0,1]. Finally, this value is added to x_i to create the new synthetic sample:

$$x_{new} = x_i + (\hat{x}_i - x_i) \times \gamma \quad (6)$$

Here, γ is an n-dimensional random vector with values in the interval [0,1], and the synthetic instance x_{new} is created on the straight path connecting x_i and $\hat{x}_i$. This new sample is produced through a random interpolation process, based on a distance metric, along a straight path between the original data point and its nearest neighbor. Unlike traditional oversampling techniques that merely replicate existing samples, this method is grounded in mathematical principles and offers superior performance in data balancing.

In this study, SMOTE was implemented using the imbalanced-learn library with sampling_strategy='minority'. Default parameters were used, namely k_neighbors=5 and metric='minkowski' with p=2 (Euclidean distance). SMOTE was applied only to the training data after the train/test split to avoid any bias or leakage into the evaluation set.

e. Backpropagation and Gradient-Based Learning: During backpropagation, the parameters w^* and b^* in each layer are determined using different optimization techniques within neural network architectures. Among the common backpropagation algorithms, Gradient Descent, ADAM, and RMSprop are considered the most popular choices in the research literature. In this study, the ADAM method was selected to perform backpropagation due to its stability, high efficiency in parameter tuning, and compatibility with the programming process.

f. Accuracy Evaluation: There are several approaches for comparing the performance of classifiers. The most widely used tool for this purpose is the confusion matrix. As shown in Table 1, the confusion matrix serves as a tool for evaluating classification accuracy [6].

In this context, TP and FP stand for *True Positive* and *False Positive*, while TN and FN represent *True Negative* and *False Negative*. A variety of measures are available to assess classifier performance. The next section outlines some of the most significant and practically useful metrics:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (7)$$

Table 1. Confusion Matrix

Observed Class	Predicted: Normal	Predicted: Abnormal
Normal	Correctly Classified as Normal (TP)	Incorrectly Classified as Abnormal (FN)
Abnormal	Incorrectly Classified as Normal (FP)	Correctly Classified as Abnormal (TN)

10.57647/ijm2c.2026.1602.12

As shown in Equation (7), Accuracy is determined by dividing the total count of correct predictions by the overall number of predictions. Additional key metrics are sensitivity and specificity, which quantify the percentage of true positives and true negatives, respectively:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \quad (8)$$

$$Specificity = \frac{TN}{TN+FP} \quad (9)$$

Furthermore, the Geometric Mean (G-mean) is determined by combining sensitivity and specificity according to the following formula:

$$G - mean = \sqrt{Sensitivity \times Specificity} \quad (10)$$

3. The Proposed Method

This research integrates a SMOTE-based data balancing phase into the preprocessing framework of a Convolutional Neural Network (CNN) aimed at Control Chart Pattern Recognition (CCPR). This approach, referred to as MP-SMOTE-CNN, is an enhanced version of the traditional CNN that enables the processing of input data through multiple channels. MP-SMOTE-CNN is designed to merge features obtained from multiple channels, with each channel emphasizing a distinct aspect of the input, thus enabling the network to capture a wider variety of relevant patterns. These channels provide alternative views of the same data, helping the model gather complementary insights and improve classification performance. Architecturally, MP-SMOTE-CNN closely resembles a standard CNN, but its primary distinction lies in processing input data through multiple independent channels. In this structure, each convolutional layer is applied independently to all input channels, producing separate feature maps that are subsequently combined through feature concatenation and fed into the fully connected layer of the network. In the present work, two separate input channels are employed: the first handles raw one-dimensional time series signals, while the second processes time series images generated via recurrence plot transformation. The architecture of the model used in the experiments is illustrated in Figure 1. The proposed architecture constructed based on the Multi-Path Synthetic Minority Oversampling Technique with Convolutional Neural Network (MP-SMOTE-CNN), which is a novel framework designed to enhance CCPR under imbalanced data conditions. The architecture integrates a data balancing stage using the SMOTE with a multi-path CNN that processes input data through two independent channels, enabling robust feature extraction and classification. Below is a detailed description of the architecture's components, data flow, and operational principles, followed by an analysis of its advantages.

The architecture employs two parallel input channels to process distinct representations of the input data.

Channel 1: Raw Time-Series Data: This channel accepts one-dimensional time-series data, typically represented as a sequence of numerical values with a fixed window length. These data capture the temporal dynamics of control chart patterns, such as trends, shifts, or cyclic behaviors.

Channel 2: Recurrence Plot Images: This channel processes two-dimensional images generated through

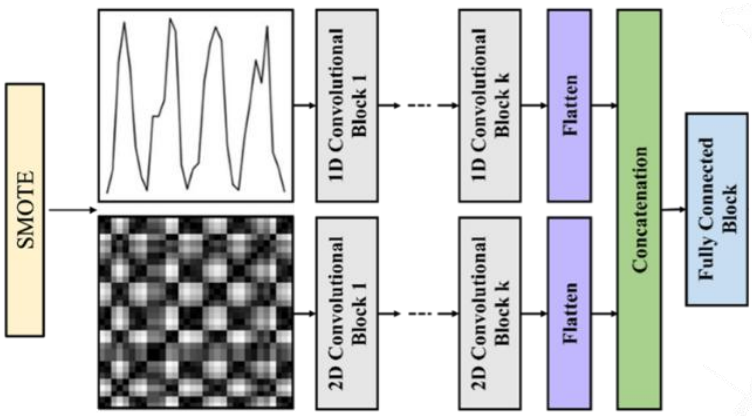

Figure 1. The architecture of the MP-SMOTE-CNN model used in the experiments

10.57647/ijm2c.2026.1602.12

recurrence plot transformation, as outlined in Section c. Recurrence plots visualize the phase space trajectory of the time-series data, encoding temporal dependencies and patterns as grayscale or binary images, which are particularly effective for capturing complex and nonlinear structures. Both channels receive input data preprocessed with SMOTE, which creates artificial data instances for the underrepresented class (abnormal patterns) through interpolation among the available samples and their k-nearest neighbors, as detailed in Section d. This step ensures a balanced dataset, mitigating bias toward the majority class (normal patterns).

Convolutional Layers: Each input channel is processed independently by a series of convolutional layers that utilize filters for capturing both spatial and temporal characteristics. As outlined in Section b, each such layer is composed of multiple filters, a bias element, and employs a Rectified Linear Unit (ReLU) as its activation mechanism. The filters in Channel 1 operate on the one-dimensional time-series data, producing feature maps that capture temporal patterns, while those in Channel 2 operate on the two-dimensional recurrence plot images, extracting spatial features.

The convolutional layers are configured with a stride parameter (s = 1), ensuring fine-grained feature extraction. The output of each convolutional layer is a matrix of reduced size, with dimensions determined by the filter size and input length (e.g., (p - q + 1) for Channel 1, where (p) is the input length and (q) is the filter length).

Pooling Layers: Following the convolutional layers, pooling layers are applied to each channel to reduce dimensionality and extract higher-level features. Max pooling is used in most stages to capture sparse features, while average pooling is applied in the final pooling layer to aggregate information. The pooling operation uses a sliding window of size (k) (typically 2 or 3) and a pooling stride. The pooling layers produce compressed feature maps, preserving the most salient features from each channel while reducing computational complexity.

Feature Concatenation: The feature maps from both channels are concatenated to form a unified feature representation. This step, central to the multi-path architecture, combines the temporal features from the raw time-series channel with the spatial features from the recurrence plot channel, enabling the model to leverage complementary information. The merged feature maps are transformed into a single-dimensional vector representation.

Fully Connected Layer and Classification: The compressed feature vector is passed through a fully connected layer, where the integrated features are processed to generate a single-dimensional output. For binary classification, a sigmoid activation function is employed to differentiate normal from abnormal control chart patterns. Optimization of the model is performed via the ADAM algorithm with backpropagation, as outlined in Section e, employing the Mean Squared Error (MSE) cost function tailored for time-series data.

Implementation Details: The architecture is implemented using Python 3.9 with Keras and TensorFlow, running on a system with an Intel i7-12900H processor (2.5 GHz), 64 GB of RAM, NVIDIA GeForce RTX 3080 Ti Laptop GPU, and a 64-bit platform. The model processes datasets with moderate (1:20) and severe (1:200) imbalance ratios, as well as the real-world Wafer dataset, demonstrating its applicability to both simulated and industrial scenarios.

The MP-SMOTE-CNN architecture offers six key advantages for CCPR, particularly in handling imbalanced data. They are:

1. Enhanced Feature Extraction through Multi-Path Processing: By employing two parallel channels, the architecture captures complementary temporal and spatial features from raw time-series and recurrence plot images, respectively. This dual-channel approach enables the model to detect complex patterns (e.g., cyclic, systematic) that single-channel CNNs may overlook, as demonstrated by the high G-mean (e.g., 0.9928 for cyclic patterns).

2. Robust Handling of Data Imbalance: The integration of SMOTE generates high-quality synthetic samples for the minority class, reducing bias toward the majority class. This is particularly effective for severe imbalance (e.g., 1:200 ratio), where the model achieved a G-mean of 0.9784–0.9928, as reported, outperforming baseline methods like Fuqua and Razzaghi [6].

3. Improved Classification Accuracy and Sensitivity: The concatenation of feature maps from both channels enhances the model's ability to generalize across diverse pattern types, leading to superior accuracy (e.g., 0.9987 on the Wafer dataset) and sensitivity (e.g., 1.0000 on the Wafer dataset). This makes MP-SMOTE-CNN highly effective for real-world industrial applications.

4. Reduced Reliance on Manual Preprocessing: The use of recurrence plots and automated feature extraction via convolutional layers minimizes the need for manual feature engineering, unlike traditional methods (e.g., decision trees). This streamlines the CCPR pipeline and enhances scalability.

5. Flexibility Across Imbalance Levels: The architecture's performance on both moderately (1:20) and severely (1:200) imbalanced datasets, as well as the real-world Wafer dataset with a 1:10 imbalance ratio, demonstrates its adaptability to varying data conditions.

6. Computational Efficiency: The use of max pooling

and a streamlined multi-path architecture reduces computational complexity while maintaining high performance, making the model suitable for real-time anomaly detection in industrial settings.

4. Results and Discussions

The performance of the proposed Multi-Path SMOTE-CNN (MP-SMOTE-CNN) model was rigorously evaluated to assess its effectiveness in control chart pattern recognition under varying degrees of data imbalance, using both simulated and real-world datasets. The experiments were conducted using Python 3.9 with Keras and TensorFlow on a system equipped with an Intel i7-12900H processor (2.5 GHz), 64 GB of RAM, and a 64-bit platform. The evaluation encompassed simulated datasets with moderate (1:20) and severe (1:200) imbalance ratios, each containing 10,000 samples, split into 75% for training and 25% for testing. Additionally, the real-world Wafer dataset, comprising 1,000 training samples and 6,164 test samples with a 1:10 imbalance ratio, was used to validate the model's performance in an industrial context. Performance was assessed using four key metrics—accuracy, sensitivity, specificity, and G-mean—with results compared against the baseline CSCNN model by Fuqua and Razzaghi [6]. Detailed results are presented in Tables 2, 3, 4, and 5, which provide comprehensive insights into the model's performance across different patterns, imbalance levels, and datasets.

The simulated datasets were generated using a mathematical model to create time-series data representing normal and abnormal control chart patterns, as defined by the equation $X_t = \mu + \epsilon_t + f(t)$, where $\mu = 0$, $\epsilon_t \sim N(0,1)$, and $f(t)$ models specific abnormal patterns. For normal (in-control) patterns, $f(t) = 0$, resulting in $X_t = \epsilon_t$. Abnormal patterns included: (1) increasing/decreasing trends, modeled as $f(t) = \pm d_1$, where d_1 is the trend slope; (2) positive/negative shifts, defined as $f(t) = 0$ before the shift and $f(t) = \pm d_2$ after the shift, where d_2 is the shift magnitude; (3) cyclic patterns, expressed as $f(t) = d_3 . sin\left(\frac{2\pi t}{\omega}\right)$, with d_3 as the amplitude and $\omega = 8$ as the period; and (4) systematic patterns, given by $f(t) = d_4 . (-1)^t$ where d_4 governs the systematic variation. The imbalance ratio ρ, defined as the proportion of abnormal samples to total samples, was set to 1:20 (moderate) and 1:200 (severe), corresponding to 500 and 50 abnormal samples, respectively, out of 10,000 total samples. The window length T and abnormality parameters (d_1, d_2, d_3, d_4) were adjusted to simulate realistic control chart scenarios, as detailed in the results below.

For the moderately imbalanced simulated dataset (1:20), the MP-SMOTE-CNN model was evaluated across six control chart patterns: upward trend, downward trend, upward shift, downward shift, cyclic, and systematic, as detailed in Table 2. For the upward trend pattern $(T = 20,\ d_1 = 0.005)$, the model achieved a G-mean of 0.7134, sensitivity of 0.8216, and specificity of 0.6200, indicating strong detection of abnormal samples but relatively weaker performance for normal samples, likely due to the subtle nature of the trend. The downward trend pattern $(T = 20,\ d_1 = 0.005)$ yielded a comparable G-mean of 0.7123, sensitivity of 0.8201, and specificity of 0.6191, suggesting that trend direction had minimal impact on performance. For the upward shift pattern $(d_2 = 0.100)$, the model recorded a G-mean of 0.7290, sensitivity of 0.8257, and specificity of 0.6443, reflecting improved detection of normal samples compared to trends. The downward shift pattern $(d_2 = 0.100)$ showed a G-mean of 0.7204, sensitivity of 0.8360, and specificity of 0.6205, indicating enhanced detection of abnormal samples. The cyclic pattern $(T = 20,\ d_3 = 0.100)$ demonstrated the highest performance, with a G-mean of 0.8892, sensitivity of 0.9464, and specificity of 0.8367, showcasing an excellent balance in detecting both classes due to the distinct periodicity captured by the model's dual-channel architecture. The systematic pattern $(T = 20,\ d_4 = 0.005)$ also performed strongly, with a G-mean of 0.8515, sensitivity of 0.9335, and specificity of 0.7765, though slightly less effective than the cyclic pattern due to its complex alternating structure. These results, as shown in Table 2, highlight the model's ability to handle moderate imbalance effectively, particularly for cyclic and systematic patterns, where the dual-channel feature extraction excels. Under severe imbalance (1:200), the MP-SMOTE-CNN model demonstrated exceptional resilience across the same six patterns, with results detailed in Table 3. For the upward trend pattern $(T = 20,\ d_1 = 0.050)$, the model achieved a G-mean of 0.9784, sensitivity of 0.9973, and specificity of 0.9600, reflecting near-perfect detection of abnormal samples and strong performance for normal samples, facilitated by SMOTE's synthetic sample generation. The downward trend pattern $(T = 20,\ d_1 = 0.005)$ showed slight improvement, with a G-mean of 0.9808, sensitivity of 0.9962, and specificity of 0.9657, indicating enhanced accuracy in identifying normal samples. For the upward shift pattern $(d_2 = 0.250)$, the model recorded a G-mean of 0.9408, sensitivity of 0.9887, and specificity of 0.8952, maintaining strong detection of abnormal samples but with reduced specificity due to the pronounced shift magnitude. The downward shift pattern $(d_2 = 0.250)$ yielded a G-mean of 0.9488, sensitivity of 0.9690, and specificity of 0.9292, showing similar trends. The cyclic pattern

($T = 20,\ d_3 = 0.500$) achieved the highest performance, with a G-mean of 0.9928, sensitivity of 0.9993, and specificity of 0.9864, reflecting exceptional accuracy in detecting both classes, as the recurrence plot channel effectively captured periodic features. The systematic pattern ($T = 20,\ d_4 = 0.050$) also performed outstandingly, with a G-mean of 0.9877, sensitivity of 0.9993, and specificity of 0.9762, benefiting from the model's ability to detect alternating patterns under severe imbalance. Table 3 underscores the model's superior performance in extreme imbalance scenarios, particularly for cyclic and systematic patterns, where SMOTE and dual-channel processing significantly enhance minority class detection.

Comparative analysis with the baseline model across simulated datasets, as detailed in Table 4, further highlights the MP-SMOTE-CNN model's advancements. For the moderately imbalanced dataset (1:20), the proposed model achieved G-mean values ranging from 0.7123 (downtrend) to 0.8892 (cyclic), significantly outperforming the baseline's range of 0.4980 (uptrend/downtrend) to 0.5561 (cyclic). For instance, for the upward trend, MP-SMOTE-CNN recorded a G-mean of 0.7134 compared to the baseline's 0.4980, and for the systematic pattern, it achieved 0.8515 against 0.5486. For the severely imbalanced dataset (1:200), the proposed model's G-mean values ranged from 0.9408 (upshift) to 0.9928 (cyclic), compared to the baseline's 0.3850 (uptrend/downtrend) to 0.7077 (cyclic). Notably, the cyclic pattern saw a G-mean of 0.9928 for MP-SMOTE-CNN versus 0.7077 for the baseline, and the systematic pattern achieved 0.9877 versus 0.5205. These results demonstrate substantial improvements, particularly under severe imbalance, where the baseline model exhibited significant performance degradation due to its limited handling of minority classes.

The MP-SMOTE-CNN model was further evaluated on the real-world Wafer dataset, sourced from the UCR Time Series Classification Archive, with results compared against the baseline model by Fuqua and Razzaghi [6], as shown in Table 5. The Wafer dataset, with a 1:10 imbalance ratio, represents time-series data from semiconductor manufacturing, with 152-time steps per sample. The proposed model achieved an accuracy of 0.9987, sensitivity of 1.0000, specificity of 0.9975, and G-mean of 0.9987, demonstrating near-perfect classification performance.

Table 2. Results of Applying the MP-SMOTE-CNN model on Test Data for the Moderately Imbalanced Simulated Dataset

Pattern Type	Parameters T = 20	Sensitivity	Specificity	G-mean	Architecture
Uptrend	$d_1 = 0.005$	0.8216	0.6200	0.7134	path_1 = Conv1D(80, 2, relu) - MaxPool1D(4, 3) - Conv1D(60, 3, relu) - GlobalAveragePooling1D() - Dropout(0.15) path_2 = Conv2D(80, 2, relu) - MaxPool2D(4, 3) - Conv2D(60, 3, relu) - GlobalAveragePooling1D() - Dropout(0.15) Concatenate([path_1, path_2]) - Dense(1, sigmoid)
Downtrend	$d_1 = 0.005$	0.8201	0.6191	0.7123	
Upshift	$d_2 = 0.100$	0.8257	0.6443	0.7290	
Downshift	$d_2 = 0.100$	0.8360	0.6219	0.7204	
Cyclical	$d_3 = 0.100$	0.9464	0.8361	0.8892	path_1 = Conv1D(80, 2, relu) - MaxPool1D(4, 3) - Conv1D(60, 3, relu) - Conv1D(100, 3, relu) - Conv1D(60, 2, relu) - GlobalAveragePooling1D() - Dropout(0.15) path_2 = Conv2D(80, 2, relu) - MaxPool2D(4, 3) - Conv2D(60, 3, relu) - Conv2D(100, 3, relu) - Conv2D(60, 2, relu) - GlobalAveragePooling1D() - Dropout(0.15) Concatenate([path_1, path_2]) - Dense(1, sigmoid)
Systematic	$d_4 = 0.005$	0.9335	0.7777	0.8515	path_1 = Conv1D(80, 3, relu) - Conv1D(60, 2, relu) – MaxPool1D(2, 2) - Dropout(0.15) - Conv1D(40, 2, relu) - Conv1D(20, 3, relu) - Dropout(0.15) - GlobalAveragePooling1D() path_2 = Conv2D(80, 3, relu) - Conv2D(60, 2, relu) – MaxPool2D(2, 2) - Dropout(0.15) - Conv2D(40, 2, relu) - Conv2D(20, 3, relu) - Dropout(0.15) - GlobalAveragePooling2D() Concatenate([path_1, path_2]) - Dense(1, sigmoid)

10.57647/ijm2c.2026.1602.12

Table 3. Results of Applying the MP-SMOTE-CNN model on the Test Data for the Severely Imbalanced Simulated dataset

Pattern Type	Parameters T = 20	Sensitivity	Specificity	G-mean	Architecture
Uptrend	$d_1 = 0.050$	0.9973	0.9599	0.9784	path_1 = Conv1D(80, 2, relu) - MaxPool1D(4, 3) - Conv1D(60, 3, relu) - GlobalAveragePooling1D() - Dropout(0.15) path_2 = Conv2D(80, 2, relu) - MaxPool2D(4, 3) - Conv2D(60, 3, relu) - GlobalAveragePooling1D() - Dropout(0.15) Concatenate([path_1, path_2]) - Dense(1, sigmoid)
Downtrend	$d_1 = 0.050$	0.9961	0.9657	0.9808	
Upshift	$d_2 = 0.250$	0.9887	0.8952	0.9408	
Downshift	$d_2 = 0.250$	0.9690	0.9292	0.9488	
Cyclical	$d_3 = 0.500$	0.9993	0.9864	0.9928	path_1 = Conv1D(80, 2, relu) - MaxPool1D(4, 3) - Conv1D(60, 3, relu) - Conv1D(100, 3, relu) - Conv1D(60, 2, relu) - GlobalAveragePooling1D() - Dropout(0.15) path_2 = Conv2D(80, 2, relu) - MaxPool2D(4, 3) - Conv2D(60, 3, relu) - Conv2D(100, 3, relu) - Conv2D(60, 2, relu) - GlobalAveragePooling1D() - Dropout(0.15) Concatenate([path_1, path_2]) - Dense(1, sigmoid)
Systematic	$d_4 = 0.050$	0.9993	0.9763	0.9877	path_1 = Conv1D(80, 3, relu) - Conv1D(60, 2, relu) – MaxPool1D(2, 2) - Dropout(0.15) - Conv1D(40, 2, relu) - Conv1D(20, 3, relu) - Dropout(0.15) - GlobalAveragePooling1D() path_2 = Conv2D(80, 3, relu) - Conv2D(60, 2, relu) – MaxPool2D(2, 2) - Dropout(0.15) - Conv2D(40, 2, relu) - Conv2D(20, 3, relu) - Dropout(0.15) - GlobalAveragePooling2D() Concatenate([path_1, path_2]) - Dense(1, sigmoid)

Table 4. Comparison of the G-mean Results of the Trained MP-SMOTE-CNN Model with the Study by Fuqua and Razzaghi [6]

Type of data	Pattern type	MP-SMOTE-CNN	Fuqua & Razzaghi [6]
Simulated relatively imbalanced dataset	Uptrend	0.7134	0.4980
	Downtrend	0.7123	0.4980
	Upshift	0.7290	0.5296
	Downshift	0.7204	0.5296
	Cyclical	0.8892	0.5561
	Systematic	0.8515	0.5486
Simulated severely imbalanced dataset	Uptrend	0.9784	0.3850
	Downtrend	0.9808	0.3850
	Upshift	0.9408	0.4940
	Downshift	0.9488	0.4940
	Cyclical	0.9928	0.7077
	Systematic	0.9877	0.5205

The perfect sensitivity score indicates that the model detected all abnormal samples, critical for industrial applications where missing anomalies can lead to significant quality issues. In contrast, the baseline model recorded an accuracy of 0.9969, sensitivity of 0.9850, specificity of 0.9984, and G-mean of 0.9917, showing slightly lower performance, particularly in sensitivity and G-mean. The baseline's marginally higher specificity (0.9984 vs. 0.9975) suggests a slight advantage in identifying normal samples, but the MP-SMOTE-CNN's superior overall balance, as reflected by the G-mean, underscores its effectiveness in real-world settings.

To contextualize practicality, we report average end-to-end runtime and memory footprints for training the models. On the simulated dataset, the full run completed in 248.54 s with peak RAM = 1.553 GB and peak GPU RAM = 15.52 GB. On the real Wafer dataset, the full run completed in 31.876 s with peak RAM = 5.741 GB and peak GPU RAM = 15.61 GB. These measurements

10.57647/ijm2c.2026.1602.12

correspond to the implementation and hardware environment described in Section 3 (“Implementation Details”) and indicate that the dual-path design remains tractable for real-time or near-real-time deployment scenarios.

The MP-SMOTE-CNN model’s success is attributed to its dual-channel architecture, which processes raw time-series data and recurrence plot images to extract complementary temporal and spatial features, respectively. The simulation method’s ability to model diverse patterns (trends, shifts, cyclic, and systematic) with specific parameters (d_1, d_2, d_3, d_4) allowed for realistic evaluation across varying imbalance levels, as reflected in Tables 2 and 3. The integration of univariate and dual-channel SMOTE generates high-quality synthetic samples for the minority class, mitigating bias toward the majority class and enhancing sensitivity for rare anomalies, as evidenced by high sensitivity scores (e.g., 0.9993 for cyclic and systematic patterns under severe imbalance). The use of max pooling and a streamlined architecture ensures computational efficiency, while automated feature extraction reduces reliance on manual feature engineering, making the model highly suitable for real-time industrial anomaly detection. The model’s flexibility across moderate and severe imbalance levels (Tables 2 and 3) and its superior performance on the Wafer dataset (Table 5) confirm its robustness and applicability in diverse industrial settings. Compared to the baseline (Table 5), the MP-SMOTE-CNN consistently achieves higher G-mean values, reflecting a better balance between sensitivity and specificity, critical for reliable CCPR.

Table 5. Comparison of the Results of Applying the Model Used in the Study on the Wafer Dataset with Baseline Metrics

Dataset	Accuracy Metrics	MP-SMOTE-CNN	Fuqua & Razzaghi [6]
Real Wafer Dataset	Accuracy	0.9987	0.9969
	Sensitivity	1.0000	0.9850
	Specificity	0.9975	0.9984
	G-mean	0.9987	0.9917
path_1 = Conv1D(80, 4, relu) – MaxPool1D(4, 4) - Conv1D(60, 3, relu) – MaxPool1D(3, 3) - Conv1D(60, 3, relu) - Conv1D(20, 2, relu) - MaxPool1D(2, 2) - GlobalAveragePooling1D() - Dropout(0.15)			
path_2 = Conv2D(80, 4, relu) – MaxPool2D(4, 4) – Conv2D(60, 3, relu) – MaxPool2D(3, 3) – Conv2D(60, 3, relu) – Conv2D(20, 2, relu) – MaxPool2D(2, 2) – GlobalAveragePooling2D() - Dropout(0.15)			
Concatenate([path_1, path_2]) - Dense(1, sigmoid)			

5. Conclusion Remarks

The Multi-Path SMOTE-CNN (MP-SMOTE-CNN) model proposed in this study represents a significant advancement in Control Chart Pattern Recognition (CCPR) for addressing the critical challenge of imbalanced data in industrial process monitoring. The novelty of the proposed technique lies in its integration of univariate and dual-channel extensions of the Synthetic Minority Over-sampling Technique (SMOTE) with a multi-path Convolutional Neural Network (CNN) architecture, which processes raw time-series data and recurrence plot images in parallel to extract complementary temporal and spatial features. This dual-channel approach, combined with SMOTE’s ability to generate high-quality synthetic minority samples, enables robust classification under both moderate (1:20) and severe (1:200) imbalance ratios, as well as in real-world industrial scenarios like the Wafer dataset (1:10 imbalance). Unlike traditional CSCNN models, such as the baseline by Fuqua and Razzaghi [6] which primarily rely on single-channel processing, the MP-SMOTE-CNN leverages diverse data representations to capture complex patterns, including trends, shifts, cyclic, and systematic anomalies, thereby enhancing detection accuracy and sensitivity for rare events.

The primary contribution of this paper is the development and empirical validation of the MP-SMOTE-CNN framework, which achieves superior performance over existing methods, as demonstrated through comprehensive experiments on simulated and real-world datasets. The results, detailed in Tables 2–5, show that the model significantly outperforms the baseline CSCNN model across various metrics. For instance, on the moderately imbalanced simulated dataset (Table 2), the model achieved G-mean values ranging from 0.7123 (downtrend) to 0.8892 (cyclic), compared to the baseline’s 0.4980–0.5561. Under severe imbalance (Table 3), the G-mean ranged from 0.9408 (upshift) to 0.9928 (cyclic), far surpassing the baseline’s 0.3850–0.7077. On the real-world Wafer dataset (Table 5), the model recorded an accuracy of 0.9987, sensitivity of 1.0000, and G-mean of 0.9987, outperforming the baseline’s 0.9969, 0.9850, and 0.9917, respectively. These results highlight the model’s ability to maintain high sensitivity and balanced performance, even in extreme imbalance scenarios, addressing a critical gap in prior research that often focused on moderately imbalanced datasets.

The advantages of the MP-SMOTE-CNN model are multifaceted, stemming from its innovative architecture and data preprocessing strategy. By processing raw

time-series and recurrence plot images through independent convolutional pathways, the model captures a richer feature set, enabling the detection of subtle and complex patterns that single-channel models may overlook. The incorporation of SMOTE ensures effective handling of data imbalance by generating synthetic minority samples that enhance model generalization without overfitting, as evidenced by the high sensitivity scores (e.g., 0.9993 for cyclic and systematic patterns under severe imbalance). The use of max pooling and average pooling in the final layer reduces computational complexity while preserving salient features, making the model suitable for real-time industrial applications. Furthermore, the automated feature extraction through convolutional layers minimizes the need for manual feature engineering, unlike traditional methods such as decision trees or support vector machines, thereby improving scalability and adaptability. The model's robustness across diverse imbalance levels and its superior performance on both simulated and real-world datasets, as shown in Tables 2–5, underscore its potential for practical deployment in industrial settings, particularly for semiconductor manufacturing and other quality-critical processes.

The superior performance of the multi-path architecture can be explained by the complementary nature of its two channels. The raw time-series path preserves sequential and fine-grained temporal variations, which are essential for detecting subtle shifts and trends. In contrast, the recurrence plot path encodes spatial structures and nonlinear dependencies in the reconstructed phase space, which are particularly effective for capturing periodicity and alternation in cyclic and systematic patterns. By integrating these two feature spaces, the model achieves higher sensitivity and balanced G-mean, as demonstrated in Tables 2 and 3.

Despite the promising results, this study has some limitations. First, the dual-path CNN increases computational complexity relative to single-path models, which may pose challenges for deployment in resource-constrained environments. Second, although SMOTE effectively addresses class imbalance, the introduction of synthetic samples can increase the risk of overfitting, particularly under extreme imbalance ratios. Future work will extend this framework along four axes: (i) architectures—integrating advanced hybrids such as Transformer-CNNs with resampling [20] and Graph Neural Networks to better capture long-range dependencies and relational structure; (ii) data & protocols—evaluating against a broader set of baselines and across larger, more diverse datasets (e.g., medical diagnostics, fraud detection, manufacturing quality control) to strengthen external validity; (iii) simulation realism—expanding the generator to include additional pattern types and variable window lengths for stress-testing robustness; and (iv) imbalance handling—systematically tuning SMOTE (e.g., k-nearest neighbors) and exploring hybrid resampling strategies beyond classical SMOTE. These directions build directly on the empirical gains documented in Tables 2–5 and aim to further improve performance under extremely imbalanced scenarios while enhancing generalizability.

Authors Contribution
All the authors have participated sufficiently in the intellectual content, conception and design of this work or the analysis and interpretation of the data (when applicable), as well as the writing of the manuscript.

Availability of data and materials
The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Conflict of interests
The author states that there is no conflict of interest.

References

[1] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263-1284. DOI: 10.1109/TKDE.2008.239

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321-357. DOI: 10.1613/jair.953

[3] Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, 61, 863-905. DOI: 10.1613/jair.1.11192

[4] Farsi, M. N., Amirfakhrian, M., & Vasiq, A. (2014). An improved controlled chaotic neural network for pattern recognition. *International Journal of Mathematical Modelling and Computers*, 4(3), 267-276.

[5] Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 249-259. DOI: 10.1016/j.neunet.2018.07.011

[6] Fuqua, D., & Razzaghi, T. (2020). A cost-sensitive convolution neural network learning for control chart pattern recognition. *Expert Systems with Applications*, 150, 113275. DOI: 10.1016/j.eswa.2020.113275

[7] Wang, S., Zhong, Z., Zhao, Y., & Zuo, L. (2021). A variational autoencoder enhanced deep learning model for wafer defect imbalanced classification. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 11(12), 2055-2060. DOI: 10.1109/TCPMT.2021.3114639

[8] Swana, E. F., Doorsamy, W., & Bokoro, P. (2022). Tomek link and SMOTE approaches for machine fault classification with an imbalanced dataset. *Sensors*, 22(9), 3246. DOI: 10.3390/s22093246

[9] Duan, F., Zhang, S., Yan, Y., & Cai, Z. (2022). An oversampling method of unbalanced data for mechanical fault diagnosis based on MeanRadius-SMOTE. *Sensors*, 22(14), 5166. DOI: 10.3390/s22145166

[10] Xue, L., Wu, H., Zheng, H., & He, Z. (2023). Control chart pattern recognition for imbalanced data based on multi-feature fusion using convolutional neural network. *Computers & Industrial Engineering*, 182, 109410. DOI: 10.1016/j.cie.2023.109410

[11] Cheng, C. S., Chen, P. W., Hsieh, Y. C., & Wu, Y. T. (2023). Multivariate process control chart pattern classification using multi-channel deep convolutional neural networks. *Mathematics*, 11(15), 3291. DOI: 10.3390/math11153291

[12] Naser, M. M. R., Varnamkhasti, M. J., Mohammed, H. J., & Aghajani, M. (2024). Artificial Intelligence as a Catalyst for Operational Excellence in Iraqi Industries: Implementation of a Proposed Model. *International Journal of Mathematical Modelling and Computers*, 14(2).

[13] Li, Y., Dai, W., & He, Y. (2024). Control chart pattern recognition under small shifts based on multi-scale weighted ordinal pattern and ensemble classifier. *Computers & Industrial Engineering*, 189, 109940. DOI: 10.1016/j.cie.2024.109940

[14] Chu, H., et al. (2024). Pattern recognition of control charts based on data feature enhancement and ensemble learning of classifiers for dimensional accuracy of products. *International Journal of Production Research*, 1-20. DOI: 10.1080/00207543.2024.2326213

[15] Xue, L., Zhu, D., Li, R., Si, S., & Wu, H. (2025). Control chart pattern recognition with variable window size for imbalanced data based on convolutional neural network with a convolutional block attention module. *Neurocomputing*, 131226. DOI: 10.1016/j.neucom.2024.131226

[16] Zan, T., Jia, X., Guo, X., Wang, M., Gao, X., & Gao, P. (2025). Research on variable-length control chart pattern recognition based on sliding window method and SECNN-BiLSTM. *Scientific Reports*, 15(1), 5921. DOI: 10.1038/s41598-025-89997-7

[17] Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., & Togneri, R. (2017). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 3573-3587. DOI: 10.1109/TNNLS.2017.2730018

[18] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90. DOI: 10.1145/3065386

[19] Murray, N., & Perronnin, F. (2014). Generalized max pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2473-2480). DOI: 10.1109/CVPR.2014.317

[20] Kamal, H., & Mashaly, M. (2024). Advanced hybrid transformer-CNN deep learning model for effective intrusion detection systems with class imbalance mitigation using resampling techniques. *Future Internet*, 16(12), 481. DOI: 10.3390/fi16120481

Appendix

A comparative analytic literature survey to tackle data imbalances

Reference	Method/Technique	Key Contribution	Relevance to CCPR
Chawla et al. [2]	Synthetic Minority Over-sampling Technique (SMOTE)	Introduced SMOTE, a resampling method generating synthetic minority samples based on k-nearest neighbors, improving classification in imbalanced datasets.	Foundational method for addressing data imbalance in CCPR, widely adopted to enhance minority class detection.
He and Garcia [1]	Review of imbalanced data learning	Provided a comprehensive survey of techniques (resampling, class weighting, ensemble learning) for handling imbalanced data in machine learning.	Established the theoretical framework for understanding data imbalance challenges in CCPR applications.
Farsi et al. [4]	Improved Controlled Chaotic Neural Network (CNN)	Proposed a novel control signal and activation function to stabilize chaotic neural networks, improving convergence speed.	Demonstrated strong pattern recognition capability, suggesting potential applicability of controlled CNNs to CCPR and similar recognition tasks.
Buda et al. [5]	CNN with imbalanced data analysis	Analyzed the impact of class imbalance on CNN performance, proposing strategies to mitigate bias in deep learning models.	Highlighted CNN vulnerabilities to imbalance, relevant for designing robust CCPR models.
Fernández et al. [3]	Comprehensive study on imbalanced data learning	Synthesized methods (resampling, ensemble, cost-sensitive learning) for imbalanced datasets, emphasizing practical applications.	Provided a broad context for applying data imbalance solutions to CCPR, including SMOTE-based approaches.
Fuqua and Razzaghi [6]	Cost-sensitive CNN (CSCNN)	Proposed a deep cost-sensitive CNN with class-weighted loss to address imbalanced CCPR.	Demonstrated robust detection of rare abnormal patterns in both simulated and real datasets under moderate and severe imbalance.
Wang et al. [7]	Variational Autoencoder + CNN (VAEDLM)	Introduced a VAE-enhanced CNN that generates wafer defect maps for improved feature learning on imbalanced data.	Achieved state-of-the-art accuracy (≥99%) on WM-811K wafer dataset and improved interpretability via saliency maps and t-SNE.
Swana et al. [8]	SMOTE + Tomek Link with classifiers	Compared NB, SVM, and KNN combined with SMOTE, Tomek Link, and hybrid resampling for fault classification.	Found SMOTE + Tomek Link with KNN yielded best precision, recall, and F1-score on imbalanced machine fault data, guiding practical condition monitoring.
Duan et al. [9]	MeanRadius-SMOTE	Proposed MeanRadius-SMOTE to reduce noisy/irrelevant synthetic samples in oversampling.	Outperformed SMOTE and LR-SMOTE on multiple imbalance scenarios, improving robustness and minority fault diagnosis accuracy in engineering applications.
Xue et al. [10]	Multi-feature Fusion CNN (MFF-CNN) + SMOTE	Proposed CNN with feature fusion combined with SMOTE for imbalanced CCP recognition.	Outperformed raw data and traditional classifiers, improving recognition of abnormal CCPs in both single and mixed patterns.
Cheng et al. [11]	Multi-channel Deep CNN (MCDCNN) with recurrence plot	Proposed a deep learning model combining 1D raw data and 2D recurrence plot features for multivariate CCPR.	Outperformed traditional methods by up to 10% in detecting multivariate non-random patterns, showing robustness under imbalanced datasets.
Naser et al. [12]	AI-based Operational Excellence Model	Proposed a 5-component AI model (data collection, processing, ML/DL algorithms, decision support, and feedback) for manufacturing.	Demonstrated significant gains in textile and food sectors, showing AI's role in predictive analytics and pattern recognition in industrial processes.
Li et al. [13]	Multi-scale Weighted Ordinal Pattern + Ensemble Classifier (WOP-EC)	Proposed OP and WOP features to capture sequential and magnitude characteristics, combined with an ensemble model.	Achieved high accuracy (94.31% small shifts; 99.88% large shifts) and reduced error rates, advancing CCPR for subtle process variations.
Chu et al. [14]	Data Feature Enhancement + Ensemble Learning (Stacking)	Combined SMOTE with smooth shift processing for feature enhancement and used a stacking ensemble of BP, ELM, PNN, and SVM.	Improved weak classifier accuracy by 30% and achieved 97.31% overall accuracy, reducing misclassification in dimensional quality control CCPR.
Xue et al. [15]	CNN with Convolutional Block	Proposed MFF-CNN-CBAM for variable window size CCPR with imbalanced data,	Achieved 99.73% accuracy, outperforming traditional CCPR

Reference	Method/Technique	Key Contribution	Relevance to CCPR
	Attention Module (CNN-CBAM) + Multi-feature Fusion	integrating SMOTE, statistical/shape features, and attention-enhanced CNN.	methods, and enabled real-time online monitoring under dynamic and imbalanced process conditions.
Zan et al. [16]	SECNN-BiLSTM + Sliding Window	Proposed variable-length CCPR using SE-attention CNN with BiLSTM and sliding window transformation.	Enabled accurate recognition of variable-length control charts; validated through simulations and cloud-edge applications, enhancing practical SPC systems.
This Study (2025)	Multi-Path SMOTE-CNN (MP-SMOTE-CNN)	Proposed a multi-path CNN with univariate and dual-channel SMOTE extensions, leveraging raw time-series and recurrence plots for enhanced classification under moderate and severe imbalance.	Advanced CCPR, addressing severe imbalance with multi-path feature extraction, achieving superior accuracy and sensitivity on simulated and real-world datasets.